Abyan Ardiatama

24060120140161

Praktikum ke-2 Machine Learning

Memuat dataset ke dalam google colab menggunakan pandas dengan memasukkan url dan nam atributnya

```
import pandas
#Memuat Dataset
url = "https://archive.ics.uci.edu/ml/machine-learning-databases/blood-transfusi
#names = ['mpg','cylinders','displacement','horsepower','weight']
dataset = pandas.read_csv(url)
```

```
print(dataset)
```

```
     Recency (months)  Frequency (times)  Monetary (c.c. blood)  \
0                   2                 50                  12500
1                   0                 13                   3250
2                   1                 16                   4000
3                   2                 20                   5000
4                   1                 24                   6000
..                ...                ...                    ...
743                23                  2                    500
744                21                  2                    500
745                23                  3                    750
746                39                  1                    250
747                72                  1                    250

     Time (months)  whether he/she donated blood in March 2007
0               98                                           1
1               28                                           1
2               35                                           1
3               45                                           1
4               77                                           0
..             ...                                         ...
743             38                                           0
744             52                                           0
745             62                                           0
746             39                                           0
747             72                                           0

[748 rows x 5 columns]
```

▾ Membagi dataset menjadi 80% data training dan 20% data validasi

```
from sklearn import*
array = dataset.values
X = array[:,0:4]
Y = array[:,4]
validation_size = 0.20
seed = 7
X_train, X_validation, Y_train, Y_validation = model_selection.train_test_split(


# Test options and evaluation metric
seed = 7
scoring = 'accuracy'
```

TT **B** *I* <> 🔗 🖼️ ⇥ ⅒ ☰ ••• ψ ☺ ▭

###Memvalidasi dataset pada 5 algoritma
decision tree, dan Logistic Regression)

Memvalidasi dataset pada 5 algoritma (KNN,gaussian,svc, decision tree, dan Logistic Regression)

```python
from sklearn.neighbors import KNeighborsClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.linear_model import LogisticRegression
from sklearn import*
# Spot Check Algorithms
models = []
models.append(('KNN', KNeighborsClassifier()))
models.append(('NB', GaussianNB()))
models.append(('SVM', SVC()))
models.append(('CART', DecisionTreeClassifier()))
models.append(('LR', LogisticRegression(solver='liblinear', multi_class='ovr')))
# evaluate each model in turn
results = []
names = []

for name, model in models:
  kfold = model_selection.KFold(n_splits=5, shuffle=True, random_state=seed)
  cv_results = model_selection.cross_val_score(model, X_train, Y_train, cv=kfold
  results.append(cv_results)
  names.append(name)
  error_score='raise'
  msg = "%s: %f (%f)" % (name, cv_results.mean(), cv_results.std())
  print(msg)
```

```
�→   KNN: 0.757521 (0.021764)
    NB: 0.769314 (0.049525)
    SVM: 0.762619 (0.028621)
    CART: 0.724090 (0.051401)
    LR: 0.777675 (0.037171)
```

▼ Meguji keakuratan metode yang memiliki akurasi perkiraan tertinggi

```python
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
# Make predictions on validation dataset
LR = LogisticRegression(solver='liblinear', multi_class='ovr')
LR.fit(X_train, Y_train)
predictions = LR.predict(X_validation)
print(accuracy_score(Y_validation, predictions))
print(confusion_matrix(Y_validation, predictions))
print(classification_report(Y_validation, predictions))
```

```
0.7533333333333333
[[107    5]
 [ 32    6]]
              precision    recall  f1-score   support

           0       0.77      0.96      0.85       112
           1       0.55      0.16      0.24        38

    accuracy                           0.75       150
   macro avg       0.66      0.56      0.55       150
weighted avg       0.71      0.75      0.70       150
```

Colab paid products - Cancel contracts here

✓   0s   completed at 19:29   ● ✕